Bubble Entropy: An Entropy Almost Free of Parameters

George Manis*, Md Aktaruzzaman, and Roberto Sassi

***Abstract*—*Objective*: A critical point in any definition of entropy is the selection of the parameters employed to obtain an estimate in practice. We propose a new definition of entropy aiming to reduce the significance of this selection. *Methods:* We call the new definition *Bubble Entropy*. Bubble Entropy is based on permutation entropy, where the vectors in the embedding space are ranked. We use the *bubble sort* algorithm for the ordering procedure and count instead the number of swaps performed for each vector. Doing so, we create a more coarse-grained distribution and then compute the entropy of this distribution. *Results:* Experimental results with both real and synthetic HRV signals showed that bubble entropy presents remarkable stability and exhibits increased descriptive and discriminating power compared to all other definitions, including the most popular ones. *Conclusion:* The definition proposed is almost free of parameters. The most common ones are the scale factor *r* and the embedding dimension *m*. In our definition, the scale factor is totally eliminated and the importance of *m* is significantly reduced. The proposed method presents increased stability and discriminating power. *Significance:* After the extensive use of some entropy measures in physiological signals, typical values for their parameters have been suggested, or at least, widely used. However, the parameters are still there, application and dataset dependent, influencing the computed value and affecting the descriptive power. Reducing their significance or eliminating them alleviates the problem, decoupling the method from the data and the application, and eliminating subjective factors.**

***Index Terms*—Bubble entropy, conditional Rényi permutation entropy, embedding dimension, entropy, HRV timeseries, permutation entropy, *r* parameter elimination.**

I. Introduction

ENTROPY has been extensively used in the analysis of HRV (Heart Rate Variability) series, and generally in the analysis of timeseries in many, diverse scientific fields. *Approximate Entropy* $(ApEn)$ [1] and *Sample Entropy* $(SampEn)$ [2] are the two most common definitions and a large number of research papers have been based on their descriptive and discriminating power.

Both methods rely on two essential parameters: the dimension m and the threshold r. The parameter m defines the dimension of the space in which we embed the given timeseries and in which we look for similar vectors (m-dimensional points). *Similar* are considered those vectors that differ less than r using the maximum norm. The selection of both m and r is essential and the results rely (sometimes significantly) on them, something that makes the existence of these parameters a critical point of both methods. A small variation of these parameters may change the entropy of the given time series drastically. So, researchers, interested in entropy, are trying to eliminate or at least make the estimation less dependent on the selection of these parameters. Towards this aim, recently Luca Citi *et al.*, have proposed *Rank Entropy* [3], which is based on the amount of shuffling that the ranks of the distances between the corresponding elements of two vectors of length m undergo when the next observation is considered. In their work, they have proposed the use of r differently than is traditionally used for $ApEn$ and $SampEn$ estimations. Instead of defining a hard threshold, e.g., 20% of standard deviation of the timeseries for checking similarity between vectors, they used r to determine the maximum rank of the set of distances that will contribute to the final entropy measure. So, we can say that they have made the estimation less dependent on r, but it still requires r.

In this paper, we focus on alternative approaches which alleviate this problem, since they estimate entropy using only the parameter m. *Permutation Entropy* [4] is one of them. A rough image of the relations between all examined methods is depicted in Fig. 1, where the solid arrows indicate *derivation* and the dashed ones *inspiration*. Even though *Permutation Entropy* was not proposed to target the elimination of r, the idea of using the ranks of the elements inside the vectors to estimate the entropy does not require a second parameter.

Based on *Permutation Entropy*, *Conditional Permutation Entropy* [5] and *Rényi Permutation Entropy* [6] were proposed, in 2014 and 2015, respectively. The first one computes a conditional entropy, while the second one uses *Rényi*, instead of *Shannon Entropy*. Our contribution starts here. We propose, as a more stable and accurate metric, the combination of the latter two. We call it *Conditional Rényi Permutation Entropy* and computes the difference of *Rényi* entropy for two successive values of m.

We move one step ahead and use additional information for defining *Bubble Entropy*. This method not only investigates the

Manuscript received October 6, 2016; revised December 31, 2016; accepted January 25, 2017. Date of publication February 6, 2017; date of current version October 18, 2017. *Asterisk indicates corresponding author.*

*G. Manis is with the Department of Computer Science and Engineering, University of Ioannina, Ioannina 45110, Greece (e-mail: manis@cs.uoi.gr).

M. Aktaruzzaman was with the Dipartimento di Informatica, Università degli Studi di Milano. He is now with the Department of Computer Science and Engineering, Islamic University Kushtia.

R. Sassi is with the Dipartimento di Informatica, Università degli Studi di Milano.

Digital Object Identifier 10.1109/TBME.2017.2664105

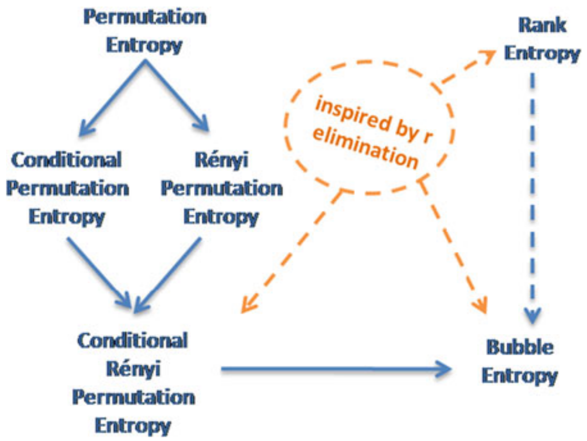

Permutation Entropy
Rank Entropy
inspired by r elimination
Conditional Permutation Entropy
Rényi Permutation Entropy
Conditional Rényi Permutation Entropy
Bubble Entropy

Fig. 1. The various definitions of entropy examined in this paper. Solid arrows indicate *derivation* and dashed ones *inspiration*.

rank of the elements inside the vectors, but also assesses the reallocation performed to sort these elements. The inspiration of this idea came from *Rank Entropy* [3], where the complexity of sorting vectors of length $m+1$ is estimated, when these vectors are already sorted according to their first m elements. We customized this idea and extended it. One of the main targets of *Rank Entropy* was the elimination of r, something that was achieved to a very good degree. With *Bubble Entropy*, we not only eliminate the necessity of r, but also minimize the importance of the second parameter m, actually suggesting a definition of entropy free of parameters.

The rest of the paper is structured as follows. We first present a bibliographic review of the field and then we describe the examined methods, both existing ones and newly suggested, in detail. Next, we present and discuss our experiments, which have been performed with both synthetic signals and real HRV recordings, in order to study the stability and the discriminating power of these methods. We compare, of course, the newly suggested methods with the rest of the methods, especially the most popular ones. A discussion section follows, while finally we outline the main outcomes of this work.

II. An Overview of Related Literature

The concept of *Permutation Entropy* (pe) was introduced by Bandt and Pompe in 2002 [4] for measuring the complexity (or regularity) of time-series. Since then, the usefulness of this metric has been examined in several fields, especially in applications for physiological signals [7]–[11]. How rapidly the popularity of this metric is increasing with time has been discussed by Zanin *et al.* in [12]. A tutorial review of the method can be found in [13]. Apart from physiological signals, *Permutation Entropy* has also been used in other application fields, including detecting speech [14], seismographic data [15] and financial time series [16].

The original concept of *Permutation Entropy* has been extended or modified in several papers [5], [17], [18]. The computation of pe depends on the embedding dimension m. Like other discrete metrics, pe might not work well for small values of m (e.g. 2), since in this case there are only a few distinct states describing the information [7]. Bandt and Pompe [4] recommended that the value of m should range from 3 to 7. However, Cao *et al.*, [7] hypothesized that a value of $m<5$ is not sufficient to detect the dynamical changes in a time-series. The larger the value of m the better the estimation of pe (it provides an upper bounds for the *Kolmogorov-Sinai Entropy* [5] for $m\rightarrow\infty$) as long as the length of the time-series is large enough [7].

Two time-series with the same ordinal structure but with different values might have the same pe [19]. To take also into consideration the values of the time-series (amplitude information) in the estimation of pe, Fadlallah *et al.*, [17] proposed a modified version of pe called *Weighted Permutation Entropy* ($WpEn$) by incorporating amplitude information when extracting the ordinal patterns. In this method, the weighted relative frequency of each motif (the pattern of m points) is calculated first and then $wpEn$ is computed. Let $Q(\pi)$ represent the amount of any permutation pattern appearing in a time-series of length N. Then, its relative frequency is defined by:

$$p(\pi)=\frac{Q(\pi)}{N-m-1} \tag{1}$$

In their study, the variance or energy of each neighboring vector was used to compute the weight. The weight w_j of some vector X_j^m with embedding dimension m was estimated as follows:

$$w_j=\frac{1}{m}\Sigma_{k=1}^{m}\left(x_{j+k-1}-\bar{X}_j^m\right)^2, \tag{2}$$

where $\bar{X}_j^m$ is the arithmetic mean of the vector X_j^m.

Vuong *et al.* [19] studied the application of $WpEn$ for recognizing different physiological states from EEG signals. According to their study, the coarse-grained symbolization of pe cannot capture properly the dynamic changes in a timeseries if no amplitude information is considered. Liu and Wang [20] have given a solution to this problem by adding an extra parameter q to take into account the difference in amplitudes of neighboring elements:

$$q=\left\lceil\frac{\max D(i)}{std(d(i)\times\alpha)}\right\rceil, \tag{3}$$

In this formula, $d(i)$ is the difference between successive elements of the given series, $D(i)$ is the difference between neighboring elements of the given vector, and α is a precision regulation factor. Next, q is added in the permutation type as an additional element. In fact, q bears information about the actual magnitude of the data in the calculation of this new metric. This new metric of entropy is termed as *Fine Grained Permutation Entropy* ($FGPE$). $FGPE$ has been shown to be more sensitive to abrupt changes in amplitude like bursts and spikes in a time-series. However, instead of reducing the number of parameter, $FGPE$ introduces an extra one (α) which also need to be calibrated.

In the definition of pe given by Bandt and Pompe [4], it was assumed that there is a rare chance of having two equal values in a sequence, and hence the case could be practically ignored. Alternatively, they proposed to rank the identical values according to their order of emergence or eliminate the equality by adding small random perturbations, such as white noise. However, for some time-series, such as HRV obtained from an ECG with a low sampling rate, there might be a large number of equal values. In that case, the Bandt and Pompe's method of dealing with equal values might fail to capture some important information hidden inside the series. For example, let us consider two vectors: $X_1 = [0.2, 0.5, 0.1, 0.4, 0.7]$ and $X_2 = [0.2, 0.5, 0.1, 0.2, 0.7]$. According to the original pe method, both vectors are mapped to the same rank vector $[3, 1, 4, 2, 5]$, even though they are different. To deal with this situation, Bian *et al.*, [18] proposed a modified version of *Permutation Entropy* ($mPeEn$). In $mPeEn$, the equal values are mapped to the unique symbol, irrespective of their order during the symbolization procedure. The vectors X_1 and X_2 are mapped to the symbolic vectors $[3, 1, 4, 2, 5]$ and $[3, 1, 1, 2, 5]$, respectively using $mPeEn$. $mPeEn$ proved empirically better when assessing complexity of HRV dynamics [18].

Bandt & Pompe defined *Sorting Entropy* along with pe in 2002, but the metric was largely overlooked until recently [5]. It is often equivalently termed *Conditional Permutation Entropy* (cpe), e.g., in [21], and we will follow this convention in here, for clarity. cpe determines the information contained in sorting the $m+1$ value among the previous m, when their order is already known (the estimate of an entropy rate and also an upper bound of the *Kolmogorov-Sinai Entropy* for $m \to \infty$):

$$\mathrm{cPE}(m) = \mathrm{PE}(m+1) - \mathrm{PE}(m). \tag{4}$$

Another definition derived from pe named *Rényi Permutation Entropy* ($RpEn$) has been introduced by Liang *et al.*, [6] in 2015. They have compared the performance of *Approximate Entropy*, *Sample Entropy*, *Fuzzy Entropy*, *Permutation Entropy*, *Tsallis Permutation Entropy* ($TpEn$), and the new $RpEn$ for monitoring the depth of anesthesia (DoA) from EEG signals analysis. The *Tsallis* and *Rényi Permutation Entropy* have been defined just based on the definition of Tsallis and Renyi entropy, respectively. If p_j represents the probability of jth permutation, then the $RpEn$ is expressed mathematically as:

$$RpEn = \frac{\log \Sigma_{j=1}^{m} p_j^{\alpha}}{(1-\alpha)\log(m)}, \tag{5}$$

where α denotes the order of *Rényi* entropy and p_j is the probability of a given ordinal pattern or rank. They found $RpEn$ as the best one among all the entropy metrics considered, when using $m = 6$ and $\alpha = 2$.

III. Methods

All methods presented in this section map a given signal onto an embedding dimension space, aiming to describe and possibly extract deeper relationships hidden in it.

Given the timeseries:

$$x = x_1, x_2, \ldots, x_N \tag{6}$$

and the size of the embedding space m, we map the signal onto the m-dimensional space:

$$X = X_1, X_2, \ldots, X_{N-m+1} \tag{7}$$

where:

$$X_i = (x_i, x_{i+1}, \ldots, x_m) \tag{8}$$

and $N_m = N - m + 1$.

Next, each X_i is sorted. Each sorted X_i is linked to a new vector J_i composed by the positions which the elements of X_i had before sorting. For example, for the vector $X_i = (6, 2, 1, 4)$ the vector J_i is $(3, 2, 4, 1)$, since the smallest element of X_i is in the third position, the second smallest one is in the second, and so on.

After having constructed the series of vectors J_i, each method follows a different way to express entropy. We will call the timeseries $J = J_1, J_2, \ldots, J_{N_m}$, as the *timeseries of indices*, for referring it easier in the following.

A. Permutation Entropy

Permutation Entropy (pe) computes the *Shannon Entropy* of the timeseries of indices. Given the timeseries J, *Permutation Entropy* first computes the probability function $p(J_i)$ of each pattern which appears in J_i. Then, *Shannon Entropy* is computed according to the definition:

$$peEn = H = -\sum_{i=1}^{n} p(J_i)\log(p(J_i)) \tag{9}$$

B. Conditional Rényi Permutation Entropy

Permutation Entropy uses the *Shannon's* definition for entropy. It is also non-conditional, since it is computed for a single value of m. In order to produce a more descriptive metric we define *Conditional Rényi Permutation Entropy* ($cRpeN$) as a combination of *Conditional Permutation Entropy* and *Rényi Permutation Entropy* described above. The name is long but descriptive enough. *Rényi* entropy is given by the formula:

$$H_\alpha(X) = \frac{1}{1-a}\log\left(\sum_{i=1}^{n} p_i^{\alpha}\right) \tag{10}$$

where $\alpha \geq 0$ and $\alpha \neq 1$. When $\alpha \to 1$ we get *Shannon Entropy* (H_1). Intuitively, we selected to use the *Rényi* entropy of order 2, since a) it gives enough emphasis on steep slopes and peaks without depressing more than necessary the lower values in the histogram and b) *Sample Entropy* which is the most popular and a widely accepted definition is also quadratic. Thus, for $\alpha = 2$ we get:

$$H_2(X) = -\log\sum_{i=1}^{n} p_i^2 \tag{11}$$

and $cRpeN$ is given by the following formula:

$$cRpeN = (H_2^{m+1} - H_2^{m})/\log(m+1) \tag{12}$$

The factor $1/\log(m+1)$ is used for normalization purposes and is the result of the subtraction of the maximum entropy for m, which is $\log(m!)$, from the maximum entropy for $m+1$, which is $\log[(m+1)!]$. Thus, after normalization, $cRpeN$ is maximized for white noise.

C. Bubble Entropy

Bubble Entropy is based on the application of a metric on the permutation procedure estimating the effort required by the permutation process. Vectors requiring the same sorting effort are grouped together and the conditional *Rényi* entropy is computed based on the new vector groups. With this procedure we reduce the number of unique possible states and create a more coarse-grained distribution based on inherent correlations.

The measure applied is the number of steps of a sorting algorithm. We use *bubble sort* and count the number of swaps necessary to sort the vector in ascending order. We will name this entropy *Bubble Entropy* and give it the acronym bEn. We calculate the number of swaps required to sort each vector and then compute the conditional *Rényi* entropy of this distribution (H^{m}_{swaps}).

bEn is expressed by the formula:

$$bEn = (H^{m+1}_{swaps} - H^{m}_{swaps}) / \log\left({}^{m+1}/_{m-1}\right) \tag{13}$$

The normalization factor is given by the difference of the maximum entropy which is $\log[1 + {}^{m(m-1)}/_{2}]$ for embedding dimension m and $\log[1 + {}^{m(m+1)}/_{2}]$ for embedding dimension $m + 1$. In each case it expresses the number of possible unique states, since with *bubble sort* is possible to perform from 0 to ${}^{m(m-1)}/_{2}$ swaps. For simplifying the normalization factor, the state *no swaps performed* was not taken into consideration, something not important for non-small values of m.

In the following pseudo-code the computation of *Bubble Entropy* is shown in steps:

1) We sort each vector X_i of m elements in ascending order, counting the number of swaps n_i necessary
2) We make an histogram from the values of n_i and we normalized it by $N - m + 1$, to obtain the probabilities p_i (describing how likely a given number of swaps n_i is)
3) From p_i, we compute the entropy H^{m}_{swaps} according to (11)
4) We repeat steps 1 to 3 to compute H^{m+1}_{swaps} for vectors of $m + 1$ elements
5) We compute bEn according to (13)

IV. Dataset

We performed experiments with both synthetic signals and Holter recordings. The synthetic signals were produced by an autoregressive model, fed with white noise. We use synthetic signals, since for a synthetic signal we can compute the theoretical value expected to be returned by each examined method. Given the theoretical value we can evaluate how fast, reliably or in a stable way the method converges to the expected value.

We also used two databases from Physionet [22]. The first one (*chf2*) consists of beat annotation files for 29 long-term ECG recordings of subjects aged 34 to 79, with congestive heart failure (NYHA classes I, II, and III). The second one (*nsr2*) includes beat annotation files for 54 long-term ECG recordings of subjects in normal sinus rhythm. The original ECG recordings were digitized at 128 samples per second, and the beat annotations were obtained by automated analysis with manual review and correction. Due to the physiology, the two populations are expected to be effectively discriminated using HRV analysis methods, including the entropy based ones.

V. Experimental Results

A. Stability Test: A Study With Synthetic Signals

In this section we evaluate the stability of the examined methods with the assist of synthetic signals. The synthetic signals are generated by an AR (AutoRegressive) model. An AR process of order M can be expressed as

$$x[n] = -\sum_{i=1}^{M} a_i x[n-i] + w[n] \tag{14}$$

where a_i are real coefficients and $w(n)$ is a white Gaussian noise (WGN) with mean zero and variance σ^2_w. The parameters of the AR model have been fitted using recordings from *nsr* dataset, available on Physionet [22]. AR processes play a central role in the analysis of HRV, since they are routinely employed for maximum entropy estimate of the series power spectrum.

By using these synthetic series obtained from the AR model and not the initial data we decouple the problem from the specific dataset and have the opportunity to study it more generally and in a more reliable way. In addition, we can feed the AR model with white noise, an input with maximum entropy, and estimate the maximum entropy given by each examined method, something that allows us to estimate how fast each method converges and how long the minimum signal length is in order to get reliable results.

Experimental results have been collected by running 200 simulations using the Monte Carlo approach. The input to the AR model is the white noise and the output is two graphs, one showing the entropy given by each method for different signal lengths N (Fig. 2) and, the second, for increasing values of m (Fig. 4).

In Fig. 2(a) the values of entropy for all different definitions and different signal sizes have been plotted. For each definition we selected a value of m which is meaningful for the specific definition and gives good results for this method: we selected $m = 2$ and $r = 0.20$ for *Sample Entropy*, $m = 10$ for *Permutation Entropy* and *Rényi Permutation Entropy*, $m = 5$ for *Conditional Permutation Entropy* and *Conditional Rényi Permutation Entropy*, and $m = 10$ for *Bubble Entropy*. We can observe, that all methods converge for small values of N, except *Permutation Entropy* and *Rényi Permutation Entropy* which seem not to converge fast enough, even for larger values of N.

Fig. 2(b) is the same plot with Fig. 2(a), but the x-axis is in logarithmic scale. In this way it is easier to observe the behavior of the methods in short signals. The conclusion that arises here is again the same, since *Permutation Entropy* and *Rényi Permutation Entropy* seem to be continuously ascending, while all other methods have stabilized their results very fast.

Fig. 3 shows how the standard deviation of computed values changes as the length of the signal increases. The parameters have been selected to be $m = 2, r = 0.2$ for *Sample Entropy* and $m = 5$ for the rest of the methods. *Bubble Entropy* presents similar variability with *Sample Entropy*. Both methods present a higher variability than that of the rest of the methods.

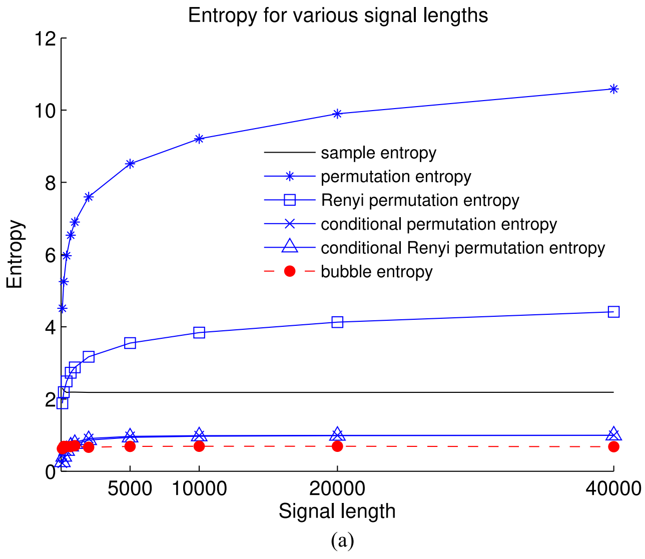


Fig. 2. Stability of computation, for increasing sizes of N. In (b) the axes are in logarithmic scale.

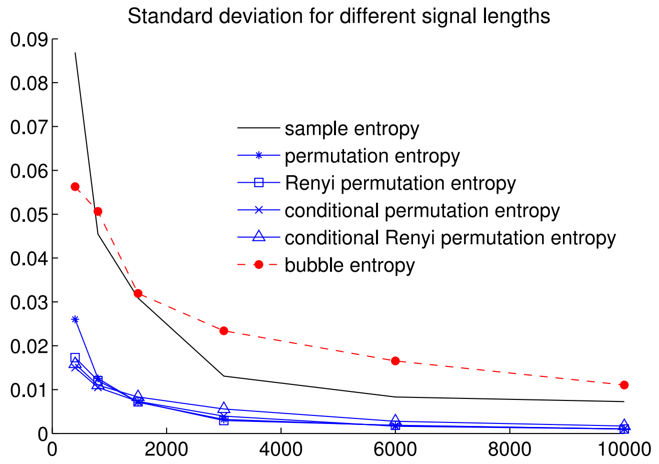


Fig. 3. Standard deviation over mean for increasing sizes of N.

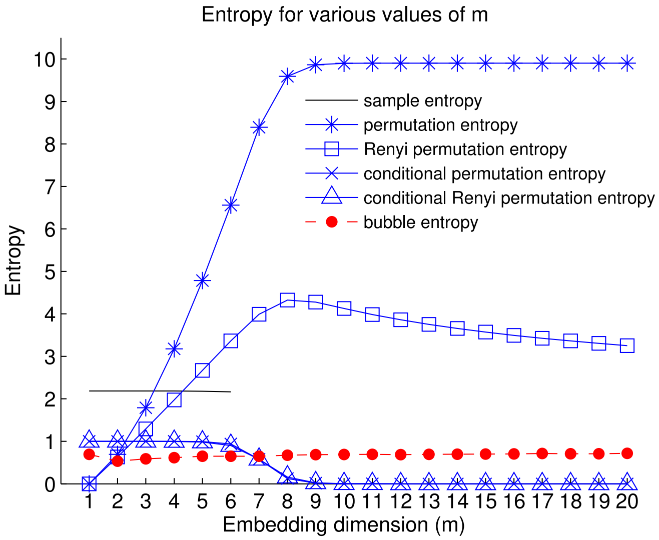
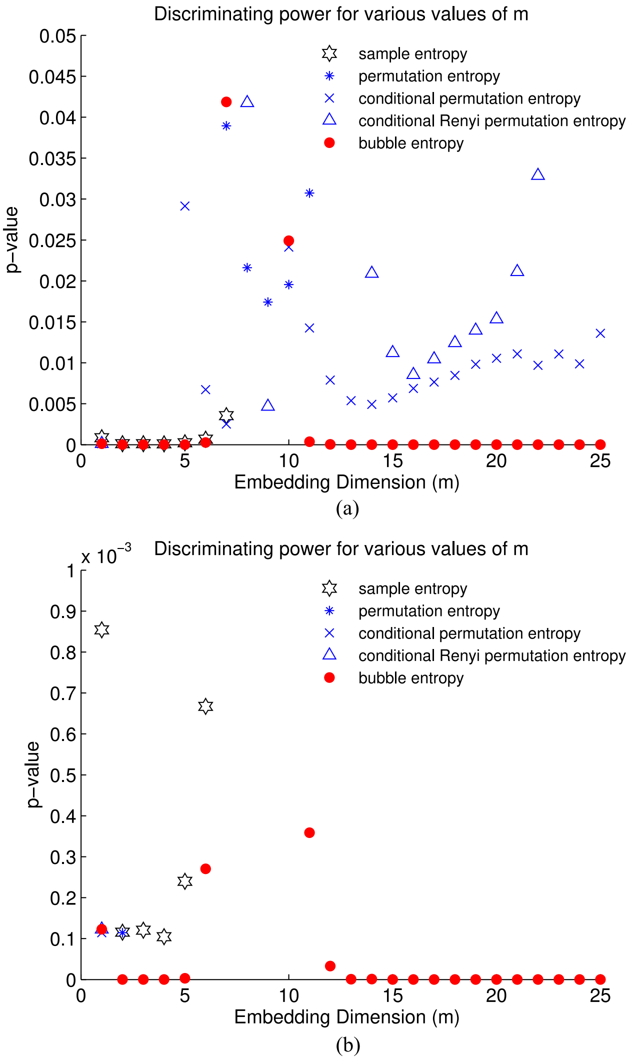


Fig. 4. Stability of computation, for increasing values of m.

In Fig. 4 we show how the values of entropy increase as m increases. We can notice that *Permutation Entropy* starts being stable after $m = 10$. *Rényi Permutation Entropy* does not converge, at least not for the examined values of m. *Conditional Permutation Entropy* and *Conditional Rényi Permutation Entropy* seem to be stable for small values of m, but, for larger ones, they give almost zero values. *Sample Entropy* presents very stable results, but for small values of m only, since it cannot be computed for larger values. *Bubble Entropy* exhibits the best behavior of all, since not only it presents a stable behavior for all values of m, but also these values seem to be independent from m, or at least not significantly affected by m. The definition of *Bubble Entropy* certainly needs the parameter m, but the selection of m does not influence significantly the computed entropy value.

Finally we tested the method using synthetic series generated with a logistic map, known to display chaotic behavior for the proper range of parameters. We generated 1000 synthetic series of different lengths ($n = 400, 800, 1600, 3200$) using a logistic map with $r = 3.9$ (the map is in the chaotic regime, as in [3]). We computed *Bubble Entropy* for each of these series and then surrogated them using $iAAFT$ (*Iterative Amplitude Adjusted Fourier Transform* with 5 iterations)[23]. We computed *Bubble Entropy* on the surrogate data and then verified that they were statistically different from the ones computed on the original series ($p << 0.01$, for $1 \leq m \leq 25$). The same experiment was performed with analogous results for *Permutation Entropy*, *Rényi Permutation Entropy* and *Conditional Rényi Permutation Entropy*.

Stability is a very important and useful property. However, it is necessary to validate how efficiently each method can discriminate groups of subjects which are expected to present statistically different entropy values. We will do this in the following section.



(a)

(b)

Fig. 5. Discriminating power of *Bubble Entropy* (a) compared to other definitions and (b) in a logarithmic scale focusing on the very low p-values.

B. Discriminating Power: Controls and Congestive Heart Failure Patients

In this section we will compare the discriminating power of each examined method using two databases, one containing recordings of heart failure patients and one with controls. We will also compare them with the most common method for entropy estimation: the *Sample Entropy*.

In Fig. 5(a) we examine p-values (t-test, null hypothesis: "means are equal") reflecting the discrimination achieved for the two groups of subjects using all examined methods: *Permutation Entropy*, *Conditional Permutation Entropy* and *Conditional Rényi Permutation Entropy*, *Sample Entropy* (the most common one), and *Bubble Entropy* (the proposed in this paper). It is not difficult to observe that, for a significant number of values of m, p-value is lower than 0.05 for most methods, something that makes us consider the result significant.

Rényi Permutation Entropy is not represented in the figure since it exhibits the poorest discrimination, with p-values being larger than 0.05 for almost all values of m. *Permutation Entropy*, shown with (blue) stars, presents a number of statistically significant values in the range $7 \leq m \leq 11$. *Conditional Permutation Entropy*, the (blue) crosses, has a large number of appearances in this graph, most of which are smaller than those of *Permutation Entropy*. After $m = 10$ the method seems to be very stable. Lowest p-values are presented by *Conditional Rényi Permutation Entropy* which are marked with the (blue) triangles, appearing in the range $9 \leq m \leq 15$.

However, for all four methods the discrimination achieved is worse than that of *Sample Entropy*. P-values for *Sample Entropy* are marked with the (black) hexagon stars in the same plot. From all possible combinations of m and r, we chose $m = 2$ and $r = 0.2$, since (a) those are the typical values and most often used and (b) they also produced the best results in our experiments ($r = 0.05$, $r = 0.1$, $r = 0.15$, $r = 0.2$, $r = 0.25$, $r = 0.3$ were checked, combined with $m = 1$, $m = 2$, $m = 3$, $m = 4$ and $m = 5$). Please note that the values stop for $m = 7$, since *Sample Entropy* was not able to find similar points for larger values of m. Also, please recall that *Sample Entropy* is a two-parameter method, when all other four use only one parameter.

The discrimination resulting with *Bubble Entropy* is shown with the (red) bullets. Please note that the number of values being smaller than 0.05 is much larger than that of any other method. Also, note that almost all p-values are extremely low. The method exhibits a remarkably stable behavior and seems to present clearly the best discriminating power of all.

Please also observe that, after a certain value of m, the selection of m does not play an important role. This observation reinforces our previous claim that, even though *Bubble Entropy* is not independent from the selection of m, the importance of the parameter is reduced significantly.

Fig. 5(b) does not add extra information. It focuses on very low p-values and further illuminates the above conclusions from Fig. 5(a) reinforcing our claims. Please note the superiority of the method on all examined methods. In this figure the superiority on *Sample Entropy* is clearer visible.

Finally, we present our experimental results for *Bubble Entropy* also in box plots (Fig. 6). Box plots give an image, not only of the mean values, but also of the distribution of the computed values. As expected, the discrimination appears to be more clear for $m \geq 12$.

Given that *Bubble Entropy* requires only one parameter (the embedding dimension m) and that this parameter does not play an important role in the discriminating capability (the selection of any relatively large value of m can lead to clear discrimination), the importance of the method seems to be significant and further examination and application of the method in other research fields would be interesting.

VI. Issues for Discussion

In this paper, we implicitly considered a time delay of $\tau = 1$ (that is we considered all the samples in a given timeseries x). In fact, our benchmark application has been heart rate variability, a context where this specific choice is often assumed. However,

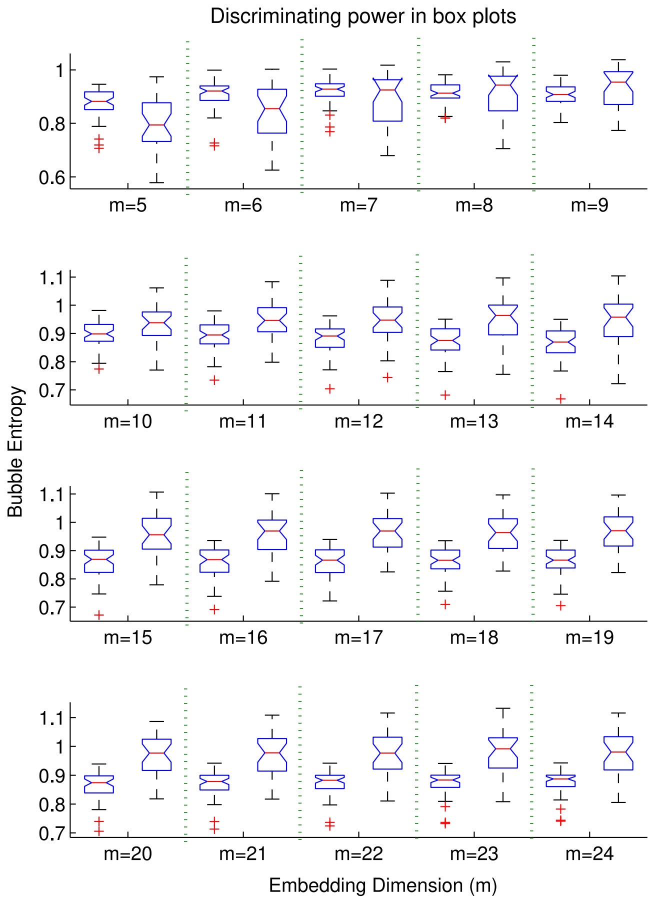


Fig. 6. Discrimination achieved shown with box plots for various values of m. For each value of m the left box plot corresponds to congestive heart failure patients and the right one to controls.

in particular when the samples x_i are derived from a continuous signal, also the sampling rate (and conversely the time delay τ) plays a role and deserves further analysis in future studies.

Another issue that is interesting to be discussed is the efficiency of the computation in terms of time complexity. *Bubble Entropy* is based on the *bubble sort* algorithm which is used for sorting each vector of the *timeseries of indices*. The algorithm complexity of *bubble sort* is $O(n^2)$, something considered expensive in terms of computational time consumption.

Sorting is generally expensive. *Quick sort* is the fastest sorting algorithm; however the steps needed to complete the sorting depends in a more arbitrary way on the input. Since the number of steps performed is important, we prefer not to use this algorithm.

However, taking a more careful look at the method, we can notice that each vector in the *timeseries of indices* differs from the preceding and succeeding vector by only one element. So we can exploit the sorted vector J_i by removing the element we do not need any more and add the new element. In this way we can produce the sorted vector J_{i+1} in only $O(m)$ time. This makes the computation of *Bubble Entropy* sufficiently fast. This is one more reason to prefer *bubble sort* and not *quick sort*.

The number of steps needed to sort a vector using *bubble sort* is the same with the number of steps required by *insertion sort*. Actually both algorithms can work here. The lastly described part with the removal of an element and the addition of a new one is simpler to be understood when thinking of *insertion sort*. The element to be removed is shifted back to the beginning of the vector, the swaps or the steps performed are counted and subtracted from the total number of steps done to sort J_i. Then the new element is inserted from the left, the number of steps are again counted until it reaches to the proper position and added to the previous difference giving the number of steps necessary to sort J_{i+1}.

In conclusion, the total complexity of the computation of *Bubble Entropy* is $O(mN)$.

VII. Conclusions

In this paper we proposed *Bubble Entropy*. *Bubble Entropy* is a new definition of entropy based on the procedure of computation of *Permutation Entropy* and presents some interesting advantages over the other definitions of entropy: a) there is no need to define an r parameter, b) after a certain value of m the discriminating power and stability of the method is not significantly dependent on m, making it an almost free of parameters method, c) it has a very stable behavior, something showed using synthesized signals generated by an autoregressive model, decoupling our conclusions from any specific dataset, and d) compared to *Sample Entropy*, *Permutation Entropy*, *Conditional Permutation Entropy*, *Rényi Permutation Entropy* and the also proposed in this paper *Conditional Rényi Permutation Entropy*, it has increased discriminating power, something exhibited by the very low p-values in the discrimination of HRV timeseries recorded from controls and congestive heart failure patients.

The elimination of r, the reduced significance of m and the encouraging experimental results make *Bubble Entropy* a very interesting method which should be further investigated and applied in biomedical signal analysis and interpretation.

References

[1] S. M. Pincus, “Approximate entropy as a measure of system complexity,” *Proc. Nat. Acad. Sci. USA*, vol. 88, no. 6, pp. 2297–2301, Mar. 1991.

[2] D. E. Lake *et al.*, “Sample entropy analysis of neonatal heart rate variability,” *Amer. J. Physiol Regul. Integr. Comp. Physiol*, vol. 283, no. 3, pp. R789–R797, Sep. 2002.

[3] L. Citi *et al.*, “Rank-based multi-scale entropy analysis of heart rate variability,” in *Proc. Comput. Cardiol. Conf.*, Sep. 2014, pp. 597–600.

[4] C. Bandt and B. Pompe, “Permutation entropy: A natural complexity measure for time series,” *Phys. Rev. Lett.*, vol. 88, no. 17 2002, Art. no. 174102.

[5] A. M. Unakafov and K. Keller, “Conditional entropy of ordinal patterns,” *Phys. D: Nonlinear Phenom.*, vol. 269, pp. 94–102, 2014.

[6] Z. Liang *et al.*, “EEG entropy measures in anesthesia,” *Front Comput. Neurosci.*, vol. 9, pp. 1–17, 2015, Art. no. 16.

[7] Y. Cao *et al.*, “Detecting dynamical changes in time series using the permutation entropy,” *Phys. Rev. E*, vol. 70, no. 4, 2004, Art. no. 046217.

[8] E. Olofsen *et al.*, “Permutation entropy of the electroencephalogram: A measure of anaesthetic drug effect,” *Br. J. Anaesthesia*, vol. 101, no. 6, pp. 810–821, 2008.

[9] A. Ravelo-Garcia, “Application of the permutation entropy over the heart rate variability for the improvement of electrocardiogram-based sleep breathing pause detection,” *Entropy*, vol. 17, no. 3, pp. 914–927, 2015.

[10] N. Mammone *et al.*, “Discovering network phenomena in the epileptic electroencephalography through permutation entropy mapping,” *Frontiers Artif. Intell. Appl.*, vol. 226, pp. 260–269, 2011.

[11] N. Nicolaou and J. Georgiou, “Detection of epileptic electroencephalogram based on permutation entropy and support vector machines,” *Expert Syst. Appl.*, vol. 39, no. 1, pp. 202–209, 2012.

[12] M. Zanin *et al.*, “Permutation entropy and its main biomedical and econophysics applications: A review,” *Entropy*, vol. 14, no. 8, pp. 1553–1577, 2012.

[13] M. Riedl *et al.*, “Practical considerations of permutation entropy: A tutorial review,” *The Eur. Phys. J. Spec. Top.*, vol. 222, no. 2, 2013, Art. no. 249262.

[14] X. L. Wu *et al.*, “Application of permutation entropy measure in detecting speech,” *Comput. Eng. Appl.*, vol. 44, no. 1, p. 240–242, 2008.

[15] C. C. Glynn and K. I. Konstantinou, “Reduction of randomness in seismic noise as a short-term precursor to a volcanic eruption,” *Sci. Rep.*, vol. 6, 2016, Art. no. 37733.

[16] L. Zunino *et al.*, “Chaotic and random dynamics discrimination in financial time series,” in *Proc. Int. Workshop Nonlinear Maps Appl.*, 2013.

[17] B. Fadlallah *et al.*, “Weighted-permutation entropy: A complexity measure for time series incorporating amplitude information,” *Phys. Rev. E Statist Nonlinear Soft Matter Phys.*, vol. 87, no. 2, 2013, Art. no. 022911.

[18] C. Bian *et al.*, “Modified permutation-entropy analysis of heartbeat dynamics,” *Phys. Rev. E*, vol. 85, no. 2, Feb. 2012, Art. no. 021906.

[19] P. L. Vuong *et al.*, “Weighted-permutation entropy as complexity measure for electroencephalographic time series of different physiological states,” in *Proc. 2014 IEEE Conf. Biomed. Eng. Sci.*, Dec. 2014, pp. 979–984.

[20] L. Xiao-Feng and W. Yue, “Fine-grained permutation entropy as a measure of natural complexity for time series,” *Chin. Phys. B*, vol. 18, no. 7, pp. 2690–2695, 2009.

[21] A. Porta *et al.*, “Limits of permutation based entropies in assessing complexity of short heart period variability,” *Physiol. Meas.*, vol. 36, pp. 755–765, 2015.

[22] A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. Ch. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley, “Physiobank, Physiotoolkit, and physionet components of a new research resource for complex physiologic signals,” *Circulation*, vol. 101, no. 23, pp. e215-e220, 2000.

[23] T. Schreiber and A. Schmitz, “Improved surrogate data for nonlinearity tests,” *Phys. Rev. Lett.*, vol. 77, no. 4, pp. 635–638, 1996.

Authors’ photographs and biographies not available at the time of publication.